OIPE IAP70 AUG 06 2007 PATENT & TRADEMARK OFFICE

Appl. No. 09/729,939
Declaration of prior invention
Reply to office action of April 2, 2007

PATENT
Case No. N0080US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/729,939
Applicant	:	Rajashri Joshi et al.
Filed	:	December 5, 2000
Titled	:	Method and System for Representation of Geographical Features in a Computer-based System

DECLARATION UNDER 37 CFR 1.131

The undersigned, OLE HENRY DORUM, hereby declares that:

1. I am a co-inventor of the invention described and claimed in the above-identified patent application.

2. On July 10, 2006, I signed a DECLARATION UNDER 37 CFR 1.131 along with co-inventors, RAJASHRI JOSHI and VIJAYA ISRANI. In that DECLARATION we indicated that before August 25, 2000, we invented a new method for representing geographic features. Part of this new method included, fitting a polynomial spline to the a geographic feature by applying a least squares approximation to data points specifying latitude and longitude coordinates to generate a plurality of control points for the polynomial spline.

3. Before August 25, 2000, a source code was developed; a redacted copy of this source code is attached hereto (Exhibit 1). The source code fits a polynomial spline to a geographic feature by applying a least squares approximation to data points specifying latitude and longitude coordinates to generate a plurality of control points for the polynomial spline. Specifically, the source code includes a function named 'uniformbsplinelsfit' that fits a polynomial spline to data points by applying a least square approximation to generate control points for the polynomial spline. Additionally, the function 'uniformbspline' plots the generated control points and the polynomial spline line.

4. Before August 25, 2000, the source code was tested and found to work. Specifically, the source code was tested and provided the output plot attached hereto (Exhibit 2). The blue points in the output plot are shape points specifying latitude and longitude coordinates of points along a road segment. The control points of the polynomial spline computed by the source code are shown as black circles in the output plot and the polynomial spline line is shown in red.

5. Exhibit 3 is a table that maps the claim elements of each independent claim onto the lines of source code of Exhibit 1 and features of the output plot of Exhibit 2.

6. All statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful statements may jeopardize the validity of the application or any patent issuing thereon.

Ole H Dorum　　　　8/2/2007

OLE HENRY DORUM　　　　Date

NAVTEQ North America, LLC
222 Merchandise Mart Plaza, Suite 900
Chicago, IL 60654
(312) 894-7000 x7365

EXHIBIT 1

Source Code

```
function [Px, Py, Kx, Ky] =
uniformbsplinelsfit(x,y,xd,yd,t,C,N,rf,m1,mc,w)

% Draws a ls fit uniform B-spline curve given the data points,
% tangent vectors at the endpoints, parameter vector, knot vector,
% number of control points, number of data points and
% regularization factor and tangent weight

for i = 1:C
   for m = 1:N
      Bv(i,m) = Bu(i-1,t(m));
   end;
end;

Bvd(1:4,1)    = [-0.5  0     0.5  0]';
Bvd(C-3:C,C)  = [ 0    -0.5  0    0.5]';
NodeWeight = 10000;

% Compute S, Sx, Sy
S = zeros(C,C); Sx = zeros(C,1); Sy = zeros(C,1);
for l = 1:C
   for i = 1:C
      for m = 1:N
            S(l,i) = S(l,i) + [redacted]
      end
      % Add extra weight to nodes %
      S(l,i) = S(l,i) + NodeWeight [redacted]
     S(l,i) = S(l,i) + NodeWeight [redacted]

      % Set up least squares equation matrix
      S(l,i) = S(l,i) + ...
[redacted]
   end
end
for l = 1:C
   for m = 1:N
      Sx(l) = Sx(l) [redacted]
      Sy(l) = Sy(l) [redacted]
   end
   % Add extra weight to nodes %
   Sx(l) = Sx(l) + NodeWeight [redacted]
   Sy(l) = Sy(l) + NodeWeight [redacted]
   Sx(l) = Sx(l) + NodeWeight [redacted]
   Sy(l) = Sy(l) + NodeWeight [redacted]
   % Set up least squares equation data matrix
   Sx(l) = Sx(l) + w*[redacted]
   Sy(l) = Sy(l) + w*[redacted]
end
[redacted]
xc = inv(S)*Sx; yc = inv(S)*Sy;
```

```
[Px, Py, Kx, Ky] = uniformbspline(xc,yc);
function [Px, Py, Kx, Ky] = uniformbspline(x,y)
% Evaluates the uniform cubic B-spline curve given the
% control points

MB = (1/6) * [ -1   3  -3   1;
                3  -6   3   0;
               -3   0   3   0;
                1   4   1   0];

MHA = [  2   -2   1   1;
        -3    3  -2  -1;
         0    0   1   0
         1    0   0   0];

MHB = (1/6)*[0      0     -1      1;
                   0      0      3      0;
                  -6      6     -2     -1;
                   6      0      0      0];

C = length(x); Delta = 0.1; InvDelta = 1/Delta;

Px = []; Py = []; Kx = []; Ky = []; GA = []; GB = [];

for seg=1:C-3
   if (seg == C-3) i = 0:Delta:1;
   else i = 0:Delta:1-Delta;
   end;
   xt = x(seg:seg+3);   yt = y(seg:seg+3);
   Pxi=polyval(MB*xt,i); Pyi=polyval(MB*yt,i);
   Kx = [Kx; Pxi(1)]; Ky = [Ky; Pyi(1)];
   Px = [Px; Pxi']; Py = [Py; Pyi'];

   GHA = inv(MHA)*MB*[xt yt];
   GA = [GA; GHA];
   GHB = inv(MHB)*MB*[xt yt];
   GB = [GB; GHB];
end;

Kx = [Kx; Pxi(end)]; Ky = [Ky; Pyi(end)];

% Find cubic spline coefficients, type A
if 0
      x = GA(:,1); y = GA(:,2);
      C = length(x); Delta = 0.1; InvDelta = 1/Delta;
      for seg=1:4:C
         xt = x(seg:seg+3);   yt = y(seg:seg+3);
         Pxi=polyval(MHA*xt,i); Pyi=polyval(MHA*yt,i);
         plot(Pxi, Pyi, 'g-');
      end;
end;

% Find cubic spline coefficients, type B
if 0
      x = GB(:,1); y = GB(:,2);
      C = length(x); Delta = 0.1; InvDelta = 1/Delta;
      for seg=1:4:C
         xt = x(seg:seg+3);   yt = y(seg:seg+3);
         Pxi=polyval(MHB*xt,i); Pyi=polyval(MHB*yt,i);
         plot(Pxi, Pyi, 'g-'); plot(xt(1), yt(1), 'go');
      end;
end
```

EXHIBIT 2

Sample Output from Source Code

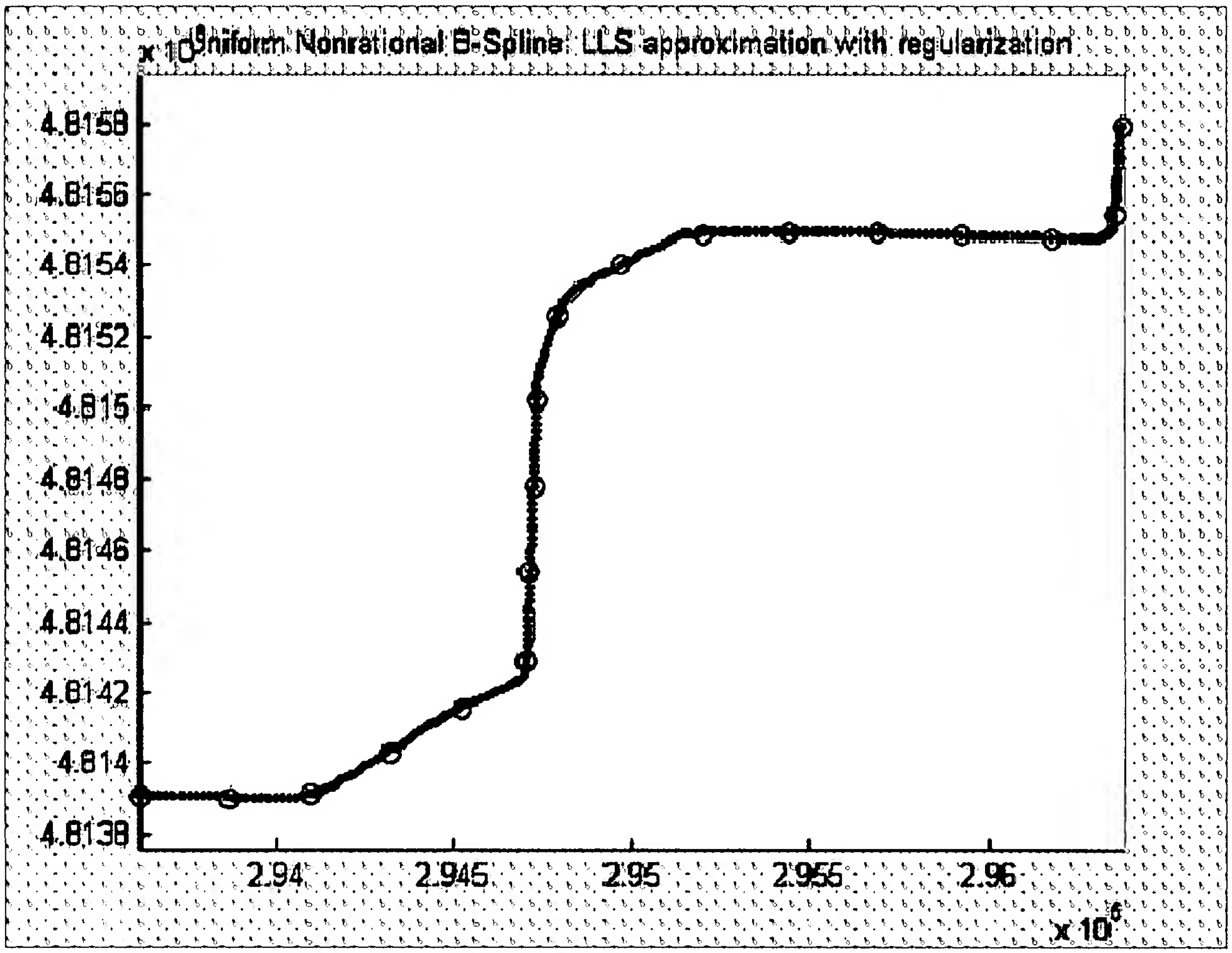

EXHIBIT 3

Table of Independent Claim Limitations that Shows Documentation for each Claim Limitation

Text of independent claims	Exhibit(s) that support claim limitations
Claim 1: A method for representing geographic features in a computer-based system, comprising: providing a first computer-usable database storing a plurality of data points specifying latitude and longitude coordinates of locations along at least one geographic feature;	Exhibit 1: page 1, lines 7-10, data points (x, y) are provided. Exhibit 2: figure shows a graph with a road represented by the plotted data points (blue dots) and x-axis and y-axis of longitude and latitude.
fitting a polynomial spline to the at least one geographic feature by applying a least squares approximation to the data points specifying latitude and longitude coordinates to generate a plurality of control points for the polynomial spline; and	Exhibit 1: page 1, lines 36-38, 51-54 and 57, control points (xc, yc) are computed using least squares equation matrix and least squares equation data matrix.
storing the control points in a second computer-usable database, the control points being usable for representing the geometry of the at least one geographic feature in the computer-based system.	Exhibit 1: page 1, line 57, control points (xc, yc). Exhibit 2: figure shows the control points (circles) plotted to represent a road.
Claim 14: A method of displaying on a computer output device a function representing a geographic feature, comprising: retrieving from a computer-usable database a plurality of spline control points associated with the geographic feature, the spline control points being derived, using a least squares approximation, from a plurality of data points specifying latitude and longitude coordinates of locations along the geographic feature;	Exhibit 1: page 1, lines 7-10, data points (x, y) are provided. Exhibit 2: figure shows a graph with a road represented by the plotted data points (blue dots) and x-axis and y-axis of longitude and latitude. Page 1, lines 36-38, 51-54 and 57, control points (xc, yc) are computed using least squares equation matrix and least squares equation data matrix.
calculating a polynomial spline using the spline control points to generate the function representing the geometry of the geographic feature; and	Exhibit 1: page 2, lines 1-40, the function uniformbspline evaluates the spline given the control points xc and yx.
displaying the function on the computer output device.	Exhibit 2: figure shows the control points (circles) and spline (red line) plotted to represent a road.

Claim 16: A method of generating a computer-usable database that represents feature geometry using a plurality of spline control points associated with a plurality of geographic features, comprising: providing a predetermined database that represents feature geometry using a plurality of data points specifying latitude and longitude coordinates of locations along the geographic features;	Exhibit 1: page 1, lines 7-10, data points (x, y) are provided. Exhibit 2: figure shows a graph with a road represented by the plotted data points (blue dots) and x-axis and y-axis of longitude and latitude.
for each of the geographic features, retrieving a corresponding set of data points specifying latitude and longitude coordinates from the predetermined database;	Exhibit 1: page 1, lines 7-10, data points (x, y) are provided.
fitting a polynomial spline to each of the geographic features by computing a plurality of control points yielding the least squares approximation to the corresponding set of data points specifying latitude and longitude coordinates; and	Exhibit 1: page 1, lines 36-38, 51-54 and 57, control points (xc, yc) are computed using least squares equation matrix and least squares equation data matrix.
storing the plurality of spline control points in the computer-usable database.	Exhibit 1: page 1, line 57, control points (xc, yc).
Claim 23: A system for displaying a function representing the geometry of a geographic feature, comprising: a database storing one or more spline control points associated with the geographic feature, the spline control points being derived, using a least squares approximation, from a plurality of data points specifying latitude and longitude coordinates of locations along the geographic feature;	Exhibit 1: page 1, lines 36-38, 51-54 and 57, control points (xc, yc) are computed using least squares equation matrix and least squares equation data matrix.
a processor configured to compute a polynomial spline using the spline control points to generate the function representing the geometry of the geographic feature; and	Exhibit 1: page 2, lines 1-40, the function uniformbspline evaluates the spline given the control points xc and yx
a display device for displaying the polyline.	Exhibit 2: figure shows the control points (circles) and spline (red line) plotted to represent a road.

Claim 29: A system for generating a plurality of spline control points that represent feature geometry, comprising: a first computer-usable database for storing a plurality of data points specifying latitude and longitude coordinates of locations along at least one geographic feature;	Exhibit 1: page 1, lines 7-10, data points (x, y) are provided. Exhibit 2: figure shows a graph with a road represented by the plotted data points (blue dots) and x-axis and y-axis of longitude and latitude.
a processor configured to apply a least squares approximation to the data points specifying latitude and longitude coordinates to generate the plurality of control points for a polynomial spline; and	Exhibit 1: page 1, lines 36-38, 51-54 and 57, control points (xc, yc) are computed using least squares equation matrix and least squares equation data matrix.
a second computer-usable database for storing the control points.	Exhibit 1: page 1, line 57, control points (xc, yc). Exhibit 2: figure shows the control points (circles) plotted to represent a road.